Hierarchical Recurrent Neural Network for Story Segmentation

Hierarchical Recurrent Neural Network for Story Segmentation

Emiru Tsunoo[1,2]*, Peter Bell*[1]*, Steve Renals*[1]

[1]The University of Edinburgh, United Kingdom
[2]Sony Corporation, Japan

Emiru.Tsunoo@jp.sony.com, Peter.Bell@ed.ac.uk, S.Renals@ed.ac.uk

Abstract

A broadcast news stream consists of a number of stories and each story consists of several sentences. We capture this structure using a hierarchical model based on a word-level Recurrent Neural Network (RNN) sentence modeling layer and a sentence-level bidirectional Long Short-Term Memory (LSTM) topic modeling layer. First, the word-level RNN layer extracts a vector embedding the sentence information from the given transcribed lexical tokens of each sentence. These sentence embedding vectors are fed into a bidirectional LSTM that models the sentence and topic transitions. A topic posterior for each sentence is estimated discriminatively and a Hidden Markov model (HMM) follows to decode the story sequence and identify story boundaries. Experiments on the topic detection and tracking (TDT2) task indicate that the hierarchical RNN topic modeling achieves the best story segmentation performance with a higher F1-measure compared to conventional state-of-the-art methods. We also compare variations of our model to infer the optimal structure for the story segmentation task.

Index Terms: spoken language processing, recurrent neural network, topic modeling, story segmentation

1. Introduction

The aim of story segmentation is to divide a sequential stream of text or audio into stories or topics. It is useful for many subsequent tasks such as summarization, topic detection, and information retrieval, and plays a crucial role for analyzing media streams. In this paper we are concerned with the segmentation of transcribed broadcast media based on a hierarchical approach in which each story consists of several sentences in a coherent order, and each sentence consists of words which are relevant to the story.

Story segmentation has been studied for decades, through various media types such as text [1, 2, 3, 4, 5, 6, 7], audio [8, 9], and video [10, 11, 12]. In the pioneering TextTiling approach [2], adjacent sentence blocks were compared using a similarity measure based on bag-of-words (BOW) or term frequency - inverted document frequency (*tf-idf*) features. Later studies indicated that globally optimized segmentation methods – such as dynamic programming (DP) and the hidden Markov model (HMM) [3, 4, 13] – can improve the performance, and usage of probabilistic topic modeling such as probabilistic latent semantic analysis (pLSA) [14, 7] and latent Dirichlet allocation (LDA) [15, 16] can further increase the accuracy. Analogous to approaches used in automatic speech recognition (ASR), deep neural networks have been combined with HMMs (DNN-HMM) and successfully applied to the story segmentation with significant improvement in performance [17]. DNNs have been also applied to similar applications including dialogue segmentation [18] and sentence boundary detection or punctuation estimation [19, 20].

Recurrent neural networks (RNNs) have extended the state of the art for general language modeling and topic/document modeling. Following the feed-forward neural prediction language model [21], Mikolov et. al. proposed using an RNN for language modelling, thus removing the limitation of finite context for predicting next words [22]. Language modelling using long short-term memory (LSTM) RNNs was proposed [23], and currently represents the state-of-the-art in language modelling [24]. To incorporate additional context, the paragraph embedding vector was introduced as an auxiliary input to an RNN language model [25, 26], and was found to improve the quality of modeling. This model factorizes into a topic factor and a word distribution for the topic, with the paragraph vector being trained to represent the topic. Hierarchical models have also been proposed for topic/document modeling [27, 28], and Lin et. al. extended the paragraph vector language model using a hierarchical RNN [29]. In this work a sentence-level RNN was used to convey an unlimited history of sentences, and by using this history vector in a similar way to a paragraph vector, each word is predicted with a word-level RNN. In an application to information retrieval, Palangi et. al. proposed an LSTM model with an output vector extracted as sentence embedding [30]. They demonstrated that specifically trained output vectors are better representations than paragraph vectors.

In this paper, we propose a hierarchical RNN for story segmentation. Each sentence is represented as a sentence embedding vector with a first word-level RNN layer, and a second sentence-level bidirectional LSTM layer models the overall story transition based on the sequence of sentence embeddings. Finally a feed-forward neural network layer predicts topic label of the input sentence, and an HMM decodes the sequence of topics and detects story boundaries. Our model is trained and evaluated on topic detection and tracking (TDT2) transcribed broadcast corpus, and compared with the state-of-the-art DNN-HMM story segmentation method [17].

2. Hierarchical Recurrent Neural Network

2.1. Overview

Broadcast news has a hierarchical character, with a top level sequence of stories, in which each story consists of multiple sentences, and each sentence consists of words which are relevant to the story. To capture this structure, we propose a hierarchical RNN model combining a sentence embedding RNN and a bidirectional LSTM story transition model. In the first layer, a word-level sentence embedding RNN, independently concentrates each sentence into a sentence embedding vector. This is followed by a second layer which models the transition of multiple stories within a chunk, for instance a program unit, using a sentence-level bidirectional RNN which considers contexts of both preceding and following sentences. The final feed-forward neural network layer estimates topic posterior

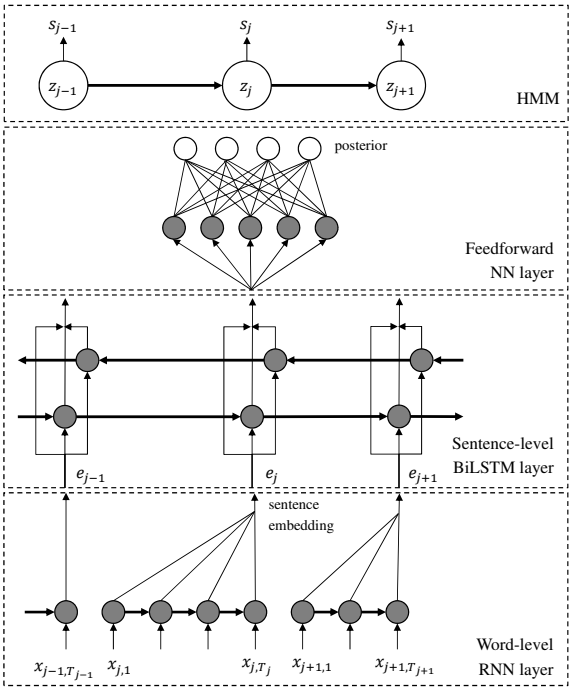

Figure 1: *Hierarchical recurrent neural network for story segmentation.*

probabilities which may be used in an HMM to decode the topic sequence, thus obtaining the story boundaries. The hierarchical RNN architecture is depicted in Figure 1.

We assume that transcriptions and sentence boundaries are available, similar to [17], as many studies regarding sentence segmentation and punctuation estimation have been done, such as [19, 20]. Given a sequence of sentences $\mathbf{s} = [s_1, ..., s_J]$ and the parameter set θ, we optimize to find the most probable topic label sequence $\hat{\mathbf{z}}$, considering all possible sequences of topic labels $\mathbf{z} = [z_1, ..., z_J]$.

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmax}}\ p(\mathbf{z}|\mathbf{s};\theta) \tag{1}$$

Analogous to the DNN-HMM acoustic model, this optimization problem can be solved with a combination of topic posterior prediction, $p(z_j|s_j)$, and transition probability modeling, $p(\mathbf{z})$, by applying Bayes' rule [31]:

$$\begin{aligned}\hat{\mathbf{z}} &= \underset{\mathbf{z}}{\operatorname{argmax}}\ p(\mathbf{s}|\mathbf{z};\theta)p(\mathbf{z})/p(\mathbf{s}) \\ &= \underset{\mathbf{z}}{\operatorname{argmax}}\ p(\mathbf{s}|\mathbf{z};\theta)p(\mathbf{z})\end{aligned} \tag{2}$$

$$p(s_j|z_j) = \frac{p(z_j|s_j)}{p(z_j)}p(s_j). \tag{3}$$

$p(\mathbf{s})$ and $p(s_j)$ do not depend to $\mathbf{z}$ and can be ignored. $p(z_j)$ is considered as prior probability, and the topic posterior $p(z_j|s_j)$ can be estimated using hierarchical RNN which we propose. The prior probability of the sequence $p(\mathbf{z})$ is modelled as the HMM transition probabilities.

2.2. Relation between Other Work

Document embeddings using paragraph vectors [25, 26] are used to augment the input to an RNN. In the case of unknown topics, the paragraph vectors must be re-trained. An alternative, reverse, approach embeds sentence information into the output vector of RNN which can be straightforwardly estimated by training discriminatively [30].

The paragraph vector approach has been extended into a hierarchical RNN, combining sentence-level and word-level RNNs [29]. In this approach, the sentence-level RNN can convey longer history while the paragraph vector is shared only within a topic or a paragraph. Our method can be considered as the reverse form of the hierarchical RNN document model, since we train it specifically for topic discrimination. This relationship is thus similar to that between the paragraph vector and the sentence embedding described above. The sentence-level RNN and word-level RNN are switched from the model in [29] for story segmentation; our sentence-level RNN uses history vector of word-level RNNs as sentence embedding vectors.

2.3. Sentence Embedding with RNN

The first layer of our model is a word-level RNN which estimates the sentence embedding vector similarly to [30]. The embedding vector concentrates information of the input sentence and represents the topic of given sentence independently. For the j-th sentence, the RNN updates the history vector $h_{j,t}$ with the given t-th word embedding vector $x_{j,t}$ within the sentence:

$$h_{j,t} = \tanh(Uh_{j,t-1} + Vx_{j,t}) \tag{4}$$

where U and V are trainable matrices. The input embedding vector $x_{j,t}$ is also to be trained. Using these history vectors, the sentence embedding vector e_j is calculated as

$$e_j = \sum_{t=1}^{T_j} \lambda_{j,t} h_{j,t} \tag{5}$$

where T_j is the total number of words in the sentence j. The weight parameters $\lambda_{j,t}$ are predefined, and they can be all set to 0 except for last word which is set to 1 to filter out only the last history vector (cf. [30]). They can be also set equally to $1/T_j$ so that the gradients spread to every time step in order to avoid the problem of vanishing or exploding gradients.

2.4. Story Transition Modeling with Bidirectional LSTM

Each story consists of multiple sentences with a coherent order. There are sometimes implicit beginning and ending notes particularly at the story changes. In order to capture the transition of sentences and stories, we adopt bidirectional LSTM, as the second layer, which has been successfully used in multiple applications such as acoustic modeling and sequence tagging [32, 33, 34]. The gated architecture of an LSTM make it possible to deal with sudden changes in the sequence, and it is reasonable to utilize LSTM for story segmentation because stories tend to change suddenly, particularly in news broadcasts. In addition, because a typical broadcast program unit contains hundreds of sentences, it is rational to adopt the LSTM which can cope with long sequence data without the vanishing/exploding gradient problem. The topic of a sentence can be represent by taking account of both side contexts of the sentence. Hence bidirectional approach is used similar to [33].

Each directional LSTM updates its parameters, for a given sentence vector e_j, and the output vector is fed into the feedforward neural network layer to estimate its topic label. We utilize LSTM with forget gate [35], and without peephole con-

nections [36]. The output vector of forward LSTM $h_{F,j}$ is computed as following.

$$
\begin{aligned}
i_{F,j} &= \sigma(W_{ei}e_j + W_{hi}h_{F,j-1} + b_i) \\
f_{F,j} &= \sigma(W_{ef}e_j + W_{hf}h_{F,j-1} + b_f) \\
c_{F,j} &= f_j \odot c_{F,j-1} + i_{F,j} \odot \tanh(W_{ec}e_j + W_{hc}h_{F,j-1} + b_c) \\
o_{F,j} &= \sigma(W_{eo}e_j + W_{ho}h_{F,j-1} + b_o) \\
h_{F,j} &= o_{F,j} \odot \tanh(c_{F,j}) \qquad (6)
\end{aligned}
$$

where σ is sigmoid function, and for the backward, parameters are calculated in the same manner. We share all the parameters W_* and b_* among forward and backward to reduce computational complexity.

2.5. Topic Posterior Prediction and HMM Decoding

The final layer computes topic posteriors sentence by sentence using a feed-forward neural network. As we want to estimate boundaries, the estimation can not rely too much on the context, otherwise the boundaries can be blurred. Therefore, in addition to the use of context information of LSTM, the sentence embedding vector e_j is used directly this layer. Let the output vectors of both forward and backward LSTM be $h_{F,j}$ and $h_{B,j}$, then the posterior $p(z_j|s_j)$ is calculated as following,

$$y_j = \sigma(W_F h_{F,j} + W_B h_{B,j} + W_r e_j + b_y) \qquad (7)$$

$$p(z_j|s_j) = g(W_p y_j + b_p) \qquad (8)$$

where g represent softmax function, and matrices W_* and bias vectors b_* are trainable.

As studies using LSTM indicate that an additional statistical model helps to improve sequential estimation [33, 34], we utilize an HMM to decode the topic sequence similar to [17].

In order to execute supervised training, the topic labels have to be given. Generally, it is easier to obtain only the boundaries of stories than the topic labels themselves. Therefore, in this paper, the labels are predefined by unsupervised clustering using CLUTO [37] similarly to [17]. Based on *tf-idf* representation, topic segments are clustered by minimizing the inter-cluster similarity and maximizing the intra-cluster similarity, then all sentences within the segments are labeled according to the clusters.

2.6. Training Procedure

The training is done jointly by minimizing cross-entropy between the target probabilities and the output posterior $p(z_j|s_j)$ using gradient descent. The target probabilities are provided according to predefined cluster labels. In order to generalize the training, the broadcast program units are broken into story segments, shuffled, and concatenated again into average program unit size. In that manner we create as many possible combination of stories as possible synthetically. The word-level RNNs are duplicated by the number of sentences in a program unit and connected in parallel to a sentence-level LSTM. The parameters are initialized with random values ranging from -0.1 to 0.1 except bias vectors, b_*, which are set to 0, and updated for every pseudo program unit. The gradients for first word-level RNN layer are clipped if their norm exceeds 0.5 to avoid the exploding gradients problem [38]. The learning rate α is set to 1 at the beginning and changed to $\alpha/2$ if the loss for validation set increases. The training process is terminated after about 30 epochs.

Table 1: *F1-measure with different number of clusters and a comparison with the other methods*

Cluster	50	100	150	170	200
TextTiling [2]			0.484		
DNN-HMM [17]	0.718	0.729	0.741	0.741	0.732
Hierarchical RNN	**0.743**	**0.739**	**0.747**	**0.744**	0.728

3. Experiments

3.1. Experimental Setup

We evaluated the hierarchical RNN on the Topic Detection and Tracking (TDT2) task [39]. We divided the data into training, validation, and test sets, 1607, 239, and 486 programs each. All words in the data were preprocessed by the Porter stemmer and stop words were removed. The total vocabulary size was 103,704.

We trained our model as described in 2.6. The hidden units of word-level RNN, sentence-level bidirectional LSTM and feed-forward neural network were all set to 256 nodes, and word embedding input vector $x_{j,t}$ was also trained with 256 dimensions. For each HMM state, the transition probability of staying same state was set to 0.8, and of switching to other state was set to the evenly divided value of remaining 0.2, as in [40, 17]. Story boundaries were detected as change points of the topic sequence decoded by the HMM, and evaluated using the F1-measure[1] comparing with the segment boundary annotation.

3.2. Story Segmentation Result

We tested our method with various numbers of clusters, from 50 to 200. We also compared with the classical approach, TextTiling [2], and the state-of-the-art method, DNN-HMM story segmentation [17], using same data set. For TextTiling, the blocks were constructed as 3 sentences for each to compare the term frequency, and this procedure did not affect the number of clusters. For the DNN-HMM, we used a context size of 60 to create BOW input features and constructed a 2-layered DNN with 256 nodes for each, similarly to [17]. The results are shown in Table 1. Overall, our method was consistently beyond the performance of DNN-HMM except for 200 clusters, and the difference between the best scores of each was statistically significant at $p < 0.05$ [41]. According to the experiment in [17], the DNN-HMM approach scored the best when the number of cluster was 170. In our replication, this was indeed the highest among the variations, however the difference was less significant than reported in [17], perhaps because the data set was not exactly the same. On the other hand, our hierarchical RNN approach had a peak at 150 clusters. The results show that our proposed model is able to represent the hierarchical topic structure effectively.

3.3. Comparison of Model Variations

We also investigated some variations of the hierarchical RNN approach. Since the RNN and LSTM are replaceable, we first evaluated the sentence embedding faculty of first word-level layer using both RNN and LSTM. Only the first RNN layer was trained, by directly calculating softmax g as following, and

[1]The F1-measure was computed with a tolerance window of 50 words according to the TDT2 standard [39].

Table 2: *Comparison of sentence embedding faculty with 150 clusters (ratios of correctly classified sentences)*

$\lambda_{j,t}$	average	last
RNN	39.60%	35.76%
LSTM	41.44%	42.29%

Table 3: *Comparison of variations of hierarchical RNN model with 150 clusters. (Bypass refers the direct usage of sentence embedding to the last feed-forward neural network discribed in section 2.5)*

Model	F1-measure
RNN-BiRNN	0.706
RNN-BiLSTM	0.729
RNN-BiLSTM-NN	0.740
RNN-BiLSTM-NN+Bypass	0.747

ratios of sentences which were correctly classified were evaluated.

$$p(z_j|s_j) = g(W'_p e_j + b'_p). \quad (9)$$

The dimensionality of embedding vector was fixed to 256. We also compared the variations of $\lambda_{j,t}$ in Equation (5), between taking "average", where all $\lambda_{j,t}$ are $1/T_j$, and filtering "last", where all $\lambda_{j,T}$ set to 0 except last one $\lambda_{j,T_j} = 1$. The result with 150 clusters was shown in Table 2 where it can be seen that taking average for calculating sentence embeddings e_j had better convergence than taking last history vector for RNN. On the other hand LSTMs were trained robustly with the variations of $\lambda_{j,t}$. Also, it indicated that although the LSTM could better represent sentences, it was not significant considering that LSTM has greater number of parameters than RNN. Therefore it was reasonable to use an RNN for the first sentence embedding layer.

Next we explored variations of our model by changing the second bidirectional LSTM layer and the last feed-forward neural network layer in the case of 150 clusters. The bidirectional LSTM layer can be easily replaced with a bidirectional RNN and we compared these approaches without using the final feed-forward layer (RNN-BiLSTM and RNN-BiRNN). We also employed the final feed-forward last neural network layer and investigated the effectiveness of bypassing the sentence embedding vector e_j to the last neural network (RNN-BiLSTM-NN and RNN-BiLSTM-NN+Bypass). The results are shown in Table 3 and indicate that, for second bidirectional layer, LSTM exceeds the performance of RNN. It also showed that the last neural network seemed to play the important role for estimating topic posterior and bypassing sentence embedding vector helped to improve the performance. We show posteriors of one validation sample for 50 clusters in Figure 2. While DNN posteriors (Figure 2-(b)) had several confusions of the topic estimation, our model without bypass (Figure 2-(c)) partly improved, and our model with bypass (Figure 2-(d)) further reduced the confusions.

4. Conclusions and Future Work

This paper proposes a hierarchical RNN approach for story segmentation task to capture the hierarchical character of broadcast news recordings. Our model uses the first RNN layer to extract a vector embedding sentence information, and uses the second layer to model the story level using a bidirectional

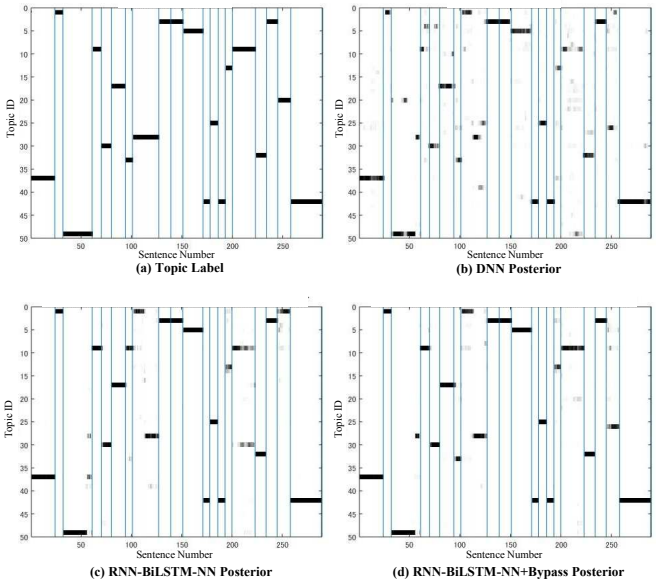

Figure 2: *Comparison of posteriors with 50 clusters. Vertical lines are story segment boundaries.*

LSTM based on the sentence embedding vector. The final neural network layer estimates the topic label according to the sentence embedding vector and both sides of context, followed by an HMM which decodes the topic sequence and obtains the boundaries. Experimentally, we have found that our hierarchical model improves on the state-of-the-art for topic segmentation in the TDT2 corpus. In addition, we compared variations of our model to explore the influence of different components in the model structure.

For future work, we are interested in combining acoustic information, since RNN has a natural character to deal with temporal modeling. It is also possible to explore using an attention mechanism to combine the history vectors to produce the sentence embedding vector.

5. Acknowledgements

This work was supported by Scalable Understanding of Multilingual MediA (SUMMA) project.

6. References

[1] M. Okumura and T. Honda, "Word sense disambiguation and text segmentation based on lexical cohesion," in *Proc. of COLING*, 1994, pp. 755–761.

[2] M. A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[3] J. Yamron, I. Carp, L. Gillick, S.Lowe, and P. van Mulvregt, "A hidden Markov model approach to text segmentation and event tracking," in *Proc. of ICASSP*, vol. 1, 1998, pp. 333–336.

[4] F. Choi, "Advances in domain independent linear text segmentation," in *Proc. of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000, pp. 26–33.

[5] M. Franz, B. Ramabhadran, T. Ward, and M. Picheny, "Automated transcription and topic segmentation of large spoken archives," in *Eurospeech*, 2003, pp. 953–956.

[6] N. Stokes, J. Carthy, and A. Smeaton, "Select: A lexical cohesion based news story segmentation system," *Journal of AI Communications*, vol. 17, no. 1, pp. 3–12, 2004.

[7] M. Lu, L. Zheng, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Broadcast news story segmentation using probabilistic latent semantic analysis and Laplacian eigenmaps," in *Porc. of APSIPA ASC*, 2011, pp. 356–360.

[8] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech in to sentences and topics," in *Speech Communication*, vol. 32, no. 1-2, 2000, pp. 127–154.

[9] A. Rosenberg and J. Hirschberg, "Story segmentation of broadcast news in English, Mandarin and Arabic," in *Proc. of HLT-NAACL*, 2006, pp. 125–128.

[10] A. Hauptmann and M. Witbrock, "Story segmentation and detection of commercials in broadcast news video," in *Proc. of Advances in Digital Libraries*, 1999, pp. 168–179.

[11] W. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar, "Discovery and fusion of salient multi-modal features towards news story segmentation," in *IS&-T/SPIE Electronic Imaging*, 2004.

[12] W. Hsu, L. Kennedy, S.-F. Chang, M. Franz, and J. R. Smith, "Columbia-IBM news video story segmentation in TRECVID 2004," Columbia University, Tech. Rep. 207-2005-3, 2005.

[13] P. Fragkou, V. Petridis, and A. Kehagias, "A dynamic programming algorithm for linear text segmentation," *Journal of Intelligent Information Systems*, vol. 23, no. 2, pp. 179–197, 2004.

[14] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc of SIGIR*, 1999, pp. 50–57.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[16] H. Misra, F. Hopfgartner, A. Goyal, P. Punitha, and J. M. Jose, "TV news story segmentation based on semantic coherence and content similarity," in *International Conference on Advances in Multimedia Modeling*, 2010, pp. 347–357.

[17] J. Yu, X. Xiao, L. Xie, E. S. Chng, and H. Li, "A DNN-HMM approach to story segmentation," in *Proc. of Interspeech*, 2016, pp. 1527–1531.

[18] Y. Song, L. Mou, R. Yan, L. Yi, Z. Zhu, X. Hu, and M. Zhang, "Dialogue session segmentation by embedding-enhanced texttiling," in *Proc. of Interspeech*, 2016, pp. 2706–2710.

[19] C. Xu, L. Xie, G. Huang, X. Xiao, E. S. Chng, and H. Li, "A deep neural network approach for sentence boundary detection in broadcast news," in *Proc. of Interspeech*, 2014, pp. 2887–2891.

[20] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *Proc. of Spoken Language Technology Workshop*, 2016.

[21] Y. Bengio, R. Duchame, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, no. 3, pp. 1137–1155, 2003.

[22] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of Interspeech*, 2010, pp. 1045–1048.

[23] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural network for language modeling," in *Proc. of Interspeech*, 2012, pp. 194–197.

[24] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazzeer, and Y. Wu, "Exploring the limits of language modeling," *ArXiv:1602.02410*, 2016.

[25] Q. Le and T. Mikolov, "Distributed representation of sentences and documents," in *Proc. of ICML*, 2014, pp. 1188–1196.

[26] A. Dai, C. Olah, and Q. Le, "Document embedding with paragraph vector," in *Proc. of NIPS 2014 in Deep Learning and Representation Learning Workshop*, 2014.

[27] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet process," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[28] J. Grimmer, "A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases," *Political Analysis*, vol. 18, no. 1, pp. 1–35, 2010.

[29] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li, "Hierarchical recurrent neural network for document modeling," in *Proc. of Empirical Methods in Natural Language Processing*, 2015, pp. 899–907.

[30] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *CoRR, abs/1502.06922*, 2015.

[31] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2015, ch. 6.

[32] A. Graves, N. Jaitly, and A. rahman Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. of ASRU*, 2013, pp. 273–278.

[33] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," *CoRP, abs/1508.01991*, 2015.

[34] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. of North American Chapter of the Association for Computational Linguistics*, 2016, pp. 260–270.

[35] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, pp. 2451–2471, 1999.

[36] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.

[37] G. Karypis, "CLUTO - a clustering toolkit," Dept. of Computer Science, University of Minnesota, Tech. Rep., 2002.

[38] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv:1211.5063v2*, 2013.

[39] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin, "NIST's 1998 topic detection and tracking evaluation (TDT2)," in *Proc. of DARPA Broadcast News Workshop*, 1999, pp. 19–24.

[40] M. Sherman and Y. Liu, "Using hidden Markov models for topic segmentation of meeting transcripts," in *Proc. of SLT*, 2008, pp. 185–188.

[41] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. of EMNLP*, 2004, pp. 388–395.